The Thesis Committee for Nithin Reddy Gopidi
Certifies that this is the approved version of the following thesis :

Emergency Response Vehicle Travel Time Analysis

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Nedialko B. Dimitrov

John Hasenbein

Emergency Response Vehicle Travel Time Analysis

by

Nithin Reddy Gopidi, B.Tech.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2016

Dedicated to my parents.

Emergency Response Vehicle Travel Time Analysis

Nithin Reddy Gopidi, M.S.E.
The University of Texas at Austin, 2016

Supervisor: Nedialko B. Dimitrov

Fire departments, ambulance services and police departments often worry if they are providing prompt response times in case of an emergency. To be effective, emergency response vehicle (ERV) have to be on the scene within a certain time of the initial emergency call. Emergency response vehicles are exempt from many traffic regulations like speed limit, crossing red signal and moreover other vehicles are expected to yield for ERV. Hence the response time analysis of ERV is very different from the regular traffic study. Advancements in the field of traffic signal control technology brought into picture new traffic signal control device (TCD). These TCDs automatically detect arrival of an ERV to turn traffic signal green for the fire engine to go through. Unfortunately, high installation costs limit the number of TCDs that can be deployed. The key goal of this article is to identify potential intersections in a traffic system for the installation of a TCDs.

In this article we propose a method of using Global Positioning System (GPS) data from ERVs to identify slow spots in the traffic system. we start with a brief overview of different map matching techniques. But, most of the map matching algorithms only try to relate the GPS points to the nearest road segment with the objective of only recreating the original path. These methods doesn't help analyze travel time. So we present Pre, Post map match process along with a customized map matching process including a nodal network of entire routes in Austin. Dynamic sizing for identify candidate points. Segment wise wait time analysis. Determine Busy intersection by frequency weighed time delay. We finally present results of the algorithm on 2 years of ERV GPS data from Austin Fire Department. Determine important intersection by frequency weighed time delay. The result can be used by different ERV to significantly improve response times, while meeting the budget restrictions.

Table of Contents

List of Figures

Chapter 1

Introduction

The public's concern for safety and improving quality of life has generated a need for improved service of numerous public-safety and transportation agencies. An important challenge in this regard is to improve transportation system management for emergency response. Minimization of emergency response time is a key focus in endeavors to improve emergency transport systems. A rapid response to an emergency situation can prevent or minimize adverse outcomes such as fatalities or the loss of property. In many cases, emergency vehicles (e.g., medical ambulances, police and security cars, fire and rescue vehicles and hazardous materials- trucks) are exempt from conventional traffic laws so that they may reach their destinations as quickly as possible; for example, they may be permitted to drive through red lights or exceed the speed limit[3].

Travel time information is important for transportation planning, route guidance as well as congestion management purposes. It is also one of the most important measures for evaluating the performance of road networks [12]. There are several software and apps currently available like Google Maps that give live updates on travel time and suggestions on best routes. But most these softwares are not equipped to handle, the special privileges that exempt ERVs from traffic regulations like speed limits or their special route requirements [2].

Emergency vehicle routing has been an attractive subject for operations researchers for many years. Most studies focused on location, fleet size, and operations performance for improving response time. The majority of these studies can be categorized into the following three groups: 1) Location of emergency vehicle stations or individual emergency vehicles within a region, subject to some performance criteria such as response time, 2) Determination of the minimum number of emergency vehicles required to cover a given area, 3) Dispatching strategies and their influence on performance results [3].

Advancements in the field of traffic signal control technology brought into picture new traffic control device (TCD). These devices automatically detect arrival of an ERV to turn a traffic signal green for the fire engine to go through. As such, a traffic light with a TCD allows for much faster travel through times for ERVs. Unfortunately, high installation costs limit the number of TCDs that can be deployed. The key goal of this article is to identify the key intersections in a traffic system for the installation of a TCD.

The main contributions of this article are: 1) Extending previous map matching methods to map match hundreds of thousands of routes for the purposes of reconstructing speed and travel time distributions across a traffic system, 2) Using the results of the first step to identify intersections, where the installation of a TCD would contribute most to reducing ERV response time. Specifically, we propose a method of using GPS data from ERVs to identify slow spots in the traffic system. The paper is organized as follows. In the next section, we formally give problem statement. This is followed by a brief literature review of map matching techniques, which are historically used for processing GPS data. But, most of the map-matching methods only try to relate the GPS points to the nearest road segment with the objective of only recreating the original path. These methods don't help reconstruct the travel time in the traffic system. We fundamentally alter these methods to be able to reconstruct speed distributions on segments of the traffic system from hundreds of thousands of GPS routes. We present results of running our method on around 2GB (approx. 300,000 paths) of Austin Fire Department (AFD) ERVs GPS data recorded from January 2013 to December 2014. We present visualizations that help to better understand traffic flow. Finally, we present a method and visualization for identifying potential intersections for installing TCDs.

Chapter 2

Literature Review: Map Matching Techniques

There has been a significant study on how GPS has established itself as a major positioning technology for various routing optimizations [10]. Typically a GPS trajectory consists of a sequence of points with latitude, longitude, and timestamp information. However, this data is not precise due to measurement errors caused by the limitation of GPS devices and sampling error caused by the sampling rate [7]. Therefore the observed GPS positions often need to be aligned with the road network on a given digital map. The process of using positioning technologies such as GPS and a road network to determine spatial reference of vehicle location is known as map matching.

Map matching is a fundamental pre-processing step for many trajectory based applications, such as moving object management, traffic flow analysis, and driving directions. The main purpose of map-matching algorithm has been to identify the correct road segment on which the vehicle is traveling and to determine the vehicle location on that segment [4, 9].

A number of map-matching algorithms have been developed by researchers around the world using different techniques such as topological analysis of spatial road network data, probabilistic theory, Kalman filter, fuzzy logic, and belief theory. Most map matching methods are generic and are applicable in a variety of situations [8], because of its wide range of applications there has been a lot of customizations for various special cases. For example, Taylor et al. [13] developed a map matching algorithm referred to as Odometer Map Matched GPS (OMMGPS) applicable to services where the most likely path or route is known in advance. Few map-matching algorithms are developed for real-time applications.

Initial approaches for map matching are geometric procedures that only take into account the distance between the GPS points and certain network elements. Like point-to-curve mapping or curve-to-curve matching based on the nearest node or nearest link approach [14]. In these methods, every two GPS points are connected by a line, i.e. curve, and the distance between this curve and the surrounding links is minimized. This approach is both easy to implement and very fast, but the main shortcoming of all geometric procedures is that they ignore the sequence of the GPS points over time as well as the connectivity of the network links. Although fast in computation, this method's performance is sensitive to the decrease of sampling frequency [11].

In contrast to the geometric procedures, topological procedures not only account for the distance between the GPS points and the network elements, but also for the sequence or history of GPS points and the connectivity of network elements. Most procedures work in two steps. First, an initial node or link is found using geometric approaches. Afterwards, choosing a link out of the set of candidate links develops the route. It does not consider any heading or speed information determined from GPS. Topological approaches consider network structure in the route generation. But, they also have draw-backs such as the potential for completely wrong route reconstruction because of an incorrect initial link computation, or parallel streets that are running closely next to each other [8].

To overcome these problems, more advanced approaches have been proposed. Advanced approaches not only take into account the whole sequence of GPS points and the network topology, but also the fact that, due to errors in the GPS measurement as well as the network coding, the nearest link or node is not necessarily the right one. Advanced map matching methods use more refined concepts such as a Kalmam Filters, Dijkstra's algorithm, or the application of Bayesian inference[8]. Our algorithm is an extension of the probabilistic map matching algorithm of Lou et al. [6], which is based on maximum likelihood reconstruction and Dijkstra's algorithm.

GPS data may come from multiple types of ERVs equipped with a variety of GPS devices. We choose to extend the algorithm of Lou et al. because it is robust to a wide range of frequency in the GPS data collection. However, their algorithm focuses on reconstructing a single route. We extend this to reconstruct hundreds of thousands of routes for the purposes of determining speed and travel time distributions across an entire traffic system.

Chapter 3

Pre Map Matching Process

To analyze total time and mean speed distributions a pre map matching process is required. Most of the map-matching algorithms only try to relate the candidate points to the nearest road segment with the objective of recreating the original path. These methods are not constructed with the goal of travel time analysis. We extend these methods to enable computing travel time distributions from ERV GPS data. Specifically, the process takes in hundreds of thousands of GPS routes and outputs travel time distributions for each road segment. The first step in the extension is to build a more refined road network, with essential attributes like speed limit.

3.1 Creating High Resolution Road Network

A refined road network creates more road segments, which allows for a more accurate representation of travel time and speed distributions (see Figure 3.1). Suppose we have a city's road network in Well Known Text (WKT) format. These road networks are generally coarse, with a minimal number of nodes required to re-create the road network.

Without the High Resolution Road Network (HRN) the closest vertex on the map for GPS point p_i would have been v_{k+1} instead of more accurate v^1_{k+1}.This difference is not a significant drawback for other map matching algorithms as long as v_{k+1} is located on the same segment. But when estimating travel time and speed distribution, more accurate representation through a HRN helps us in getting more accurate travel time distribution results. Figure 3.1 depicts the use of a HRN.

A HRN is created by adding extra nodes to the existing road network. Algorithm 1 describes a method for creating additional nodes such that distance between any two consecutive nodes on the map is less than a given threshold Th. In other words, for any consecutive vertices v_1 and v_2 on the network, if the great circle distance between these points is greater than threshold Th, we introduce α equally distributed nodes between v_1 and v_2, such that

$$\alpha = \lfloor dist(v_1, v_2)/Th \rfloor. \tag{3.1}$$

Speed limit for each edge is an essential attribute in the HRN. We need to make sure that the data source used to build the HRN has speed limit information available for each segment. Speed limit provides a base line to compare the mean speeds of ERVs and helps in identifying intersections for TCD installations.

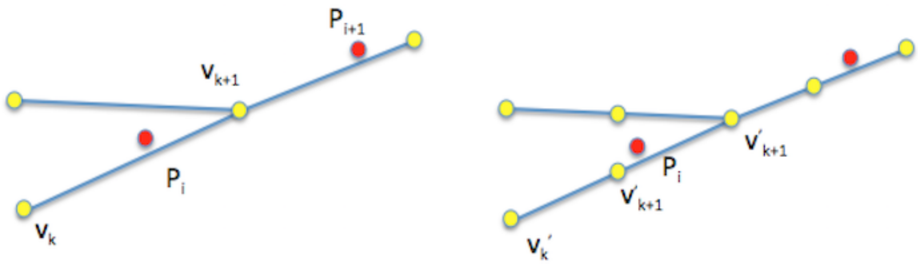

Figure 3.1: Coarse vs HRN. The left hand image is an initial coarse network. Yellow dots represent map points, and red dots represent GPS points. The right hand image is a High Resolution Road Network (HRN), where we have added additional map points through pre processing. The higher resolution map data allows for more accurate matching between the GPS points and map points.

Input: Road Network Graph G(V,E) with Vertices V and Edges E
Output: HRN Graph G'(V ',E') with Vertices V'and Edges E'

begin
  forall the $e \in E$ **do**
    if $length(e) > TH$ **then**
      `//` v_s, v_t `are vertices for edge` e
      $\alpha = \lfloor dist(v_s, v_t)/Th \rfloor$
    end
    for $i \in [0, \alpha]$ **do**
      $v'_i = v_s + ((v_t - v_s)/\alpha) * i$
    end
    $V' = V \cup [v'_0, v'_1 ...]$
    $E' = E \cup [(v_s, v'_0), (v'_0, v'_1) ...]$
  end
  Return new road Network Graph G'(V ',E')
end

Algorithm 1: Method for creating a HRN

Chapter 4

Map Matching

Existing map matching methods need modifications to facilitate travel time and speed distribution analysis. The method suggested by Lou et al. [6] focuses on converting a sequence of GPS coordinates to a sequence of map points. Our method differs mainly in candidate point selection procedure section 4.1 to map match hundreds of thousands of routes for the purposes of reconstructing speed and travel time distributions across different segments.

Map matching method presented in this article consists of 4 main components as shown in Figure 4.1, 1) Selecting candidate points, 2) Assigning spatial probability 3) Assigning transmission probability and 4) Result matching using Dijkstra's

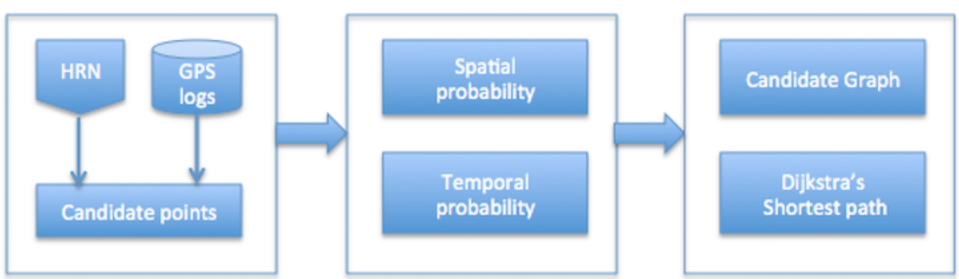

Figure 4.1: Figure showing flow chart for current map matching

Below are the definitions of few variables that we use for map matching. A GPS path is defined as collection of GPS points $L = \{p_1, p_2, ..., p_n\}$ of an ERV from its start location to an incident location. Candidate points are set of vertices on HRN that correspond to the location of the ERV when the GPS point p_i is recorded.

Selecting Candidate points: This component describes the method of selecting candidate points. Candidate points are set of vertices on HRN corresponding to a GPS point (formally defined in section 4.1). This component takes as input HRN, GPs path, a critcal limit Cl and radius R. The output is a set of candidate points for each GPS point.

Assigning Spatial probability: This component calculates likelihood that a GPS point matches a candidate point. It considers the distance between a GPS point and its candidate points to assign a spatial probability. This component requires HRN, GPS path and set of candidate points calculated in previous section to calculate spatial probability of each candidate point.

Assigning Transmission probability: This section takes into account the topological information of the road network. To avoid roundabout paths, we employ shortest path to measure the similarity between each candidate path and the true path. It estimates transmission probability between candidate points of adjacent GPS path points. This component takes as input HRN, GPS path and set of candidate points.

Result Matching using Dijkstra's: A candidate graph is constructed in this component. The nodes of the graph are the set of candidate points for each GPS observation, and the edges of the graph are between every pair of candidate points for neighboring GPS points. The nodes and edges are all assigned weight values based on spatial and transmission probability. The true path is derived by using Dijkstra's algorithm on this network.

4.1 Selecting candidate points

We follow a dynamic candidate selection process to align GPS data onto map points and calculate travel time through each path segment. For a given GPS path point p_i, all the vertices within radius R of Graph G' are selected as candidate points c_{ij} for p_i (see Figure 4.2). If the number of candidate points for p_i are less than a given critical limit Cl, we increase R with iteration. This loop continues until the number of candidate points are greater the critical limit Cl (refer algorithm 2). This dynamic selection process significantly reduces the computational time required for sections 4.3 and 4.4 of map matching method without affecting its accuracy.

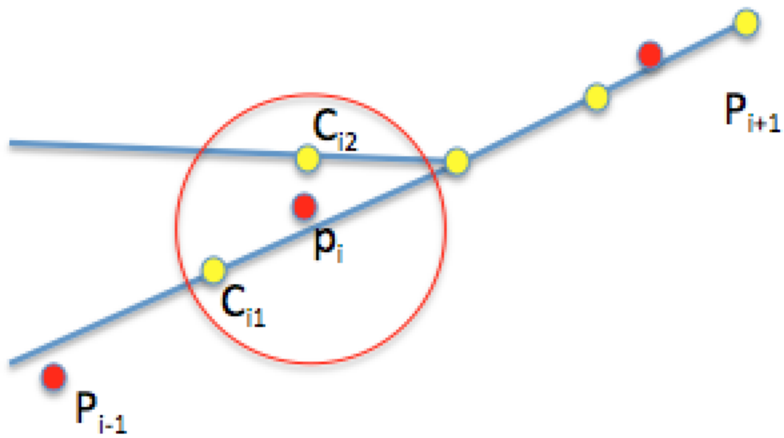

Figure 4.2: Image showing the candidate points for GPS path point p_i

Input: HRN Graph G'(V ', E') and GPS path points L
Output: Candidate points

begin
    forall the $p_i \in L = \{p_1, p_2, ..., p_n\}$ **do**
        `// Initialize` C_i `to empty array and` *iter* `to 0`
        while $Size\,(C_i) < Cl$ **do**
            $iter = iter + 1$
            $r = iter \times R$
            $C_i = \{v \in V' \mid dist(v, p_i) \leq r\}$
        end while
    end
end

Algorithm 2: Dynamic candidate point selection

4.2 Assigning Spatial Probability to Candidate points

The error in a GPS measurement can be assumed to follow Gaussian distribution $N(\mu, \sigma^2)$ that describes the distance between the recorded coordinate, p_i, and the actual locatoin c_{ij}. The spatial probability is defined as the likelihood that a GPS sampling point p_i matches a candidate point c_{ij} computed based on the distance between the two points $dist(c_{ij}, p_i)$. Formally, we define observation probability $N(c_{ij})$ of c_{ij} w.r.t. p_i as in Equation (4.1), where mean μ and distance x in Gaussian distribution are substituted by 0 and $dist(c_{ij}, p_i)$ respectively as follows

$$N(c_{ij}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(dist(c_{ij},p_i))^2/(2\sigma^2)}. \tag{4.1}$$

Spatial probability does not take into account GPS points position context. Map matching solely on this measure can lead to incorrect matches. For example, in Figure 4.2, c_{i2} will wrongly be selected instead of c_{i1} to address this issue we define transmission probability.

4.3 Assigning Transmission Probability

Transmission probability characterizes the spatial and temporal relationship between GPS points and candidate points. We define transmission probability between two candidate points $c_{i-1,t}$ and $c_{i,s}$, for two neighboring GPS path points p_{i-1} and p_i, as

$$V(c_{i-1,t} \to c_{i,s}) = \frac{dist(p_{i-1}, p_i)}{w_{(i-1,t)\to(i,s)}}, \tag{4.2}$$

where $w_{(i-1,t)\to(i,s)}$ is the length of shortest path in the HRN G' from $c_{i-1,t}$ to $c_{i,s}$, and $dist(p_{i-1}, p_i)$ is great circle distance between the GPS points. Transmission probability is intended to capture the likelihood that the "true" path from p_{i-1} to p_i follows the shortest path from $c_{i-1,t}$ to $c_{i,s}$.

Combining equations (4.1) and (4.2) we get the final likelihood that an object moves from $c_{i-1,t}$ to $c_{i,s}$ using the product of two probabilities. The following definition takes both geometric and topological information into consideration:

$$F\left(c_{i-1,t} \to c_{i,s}\right) = N\left(c_{ij}\right) * V\left(c_{i-1,t} \to c_{i,s}\right). \tag{4.3}$$

4.4 Result Matching using Dijkstra's

After assigning spatial and transmission probabilities, we generate a new candidate graph $G'_T(V'_T, E'_T)$ for GPS path $L = \{p_1, p_2, ..., p_n\}$. V'_T is a set of candidate points for each GPS path point, and E'_T is a set of edges connecting every candidate point for p_i to every candidate point for p_{i+1}, as depicted in Figure 4.3. We place a length $w_{(c_{i-1,t}\to c_{i,s})}$ given by Equation (4.4) on each edge of G'_T, where

$$w_{(c_{i-1,t}\to c_{i,s})} = -\log(F\left(c_{i-1,t} \to c_{i,s}\right)). \tag{4.4}$$

A candidate path sequence P_c for the entire trajectory L is a path in the candidate graph, denoted as $P_c : c_{1,s_1} \to c_{2,s_2} \to .....c_{n,s_n}$.The overall score for such a candidate sequence is $S(P_c) = \Sigma_{i=2}^{n} - \log(F(c_{i-1,s_{i-1}} \to c_{i,s_i}))$. From

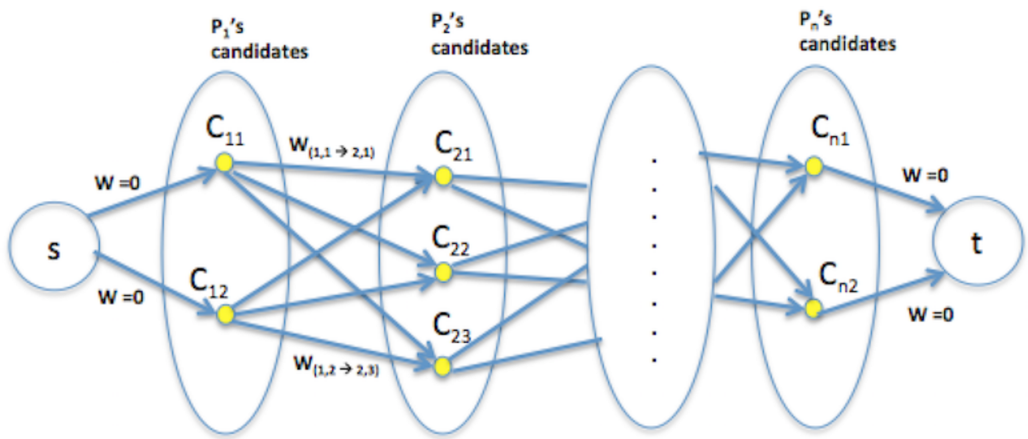

Figure 4.3: Image showing candidate graph $G'_T(V'_T, E'_T)$ created to reconstruct the best path P.

all the candidate sequences we aim to find the one with the lowest overall score as the best path for the trajectory. Additional nodes s and t are added to network G'_T with zero length edges as shown in Figure 4.3. Solve Dijkstra's shortest path from s to t with weights defined by Equation (4.4) to get the candidate sequence that best matches true path P for a given trajectory L.

Chapter 5

Post Map Matching

Map matching is a process applied to a single path of GPS coordinates. We now describe the process of taking hundreds of thousands of such paths to produce speed distributions for each segment of the HRN. For a sequence of GPS coordinates $L = \{p_1, p_2, ..., p_n\}$, we call the corresponding output of map matching as the *true path* $P = \{m_1, m_2, ..., m_n\}$. The true path contains only points from the HRN, and has timing information associated with each point from the GPS data.

5.1 Segment Travel Time and Speed Distributions

In this section, we describe how to calculate travel time and speed distributions by all ERVs on any given HRN segment. For this we need to process all the true paths resulting from map matching. The same sequence of steps is repeated for each true path. In the end, we have an empirical travel time distribution for each segment in the HRN. The distributions of travel times, along with the segment lengths, allow for computation of speed distributions across the segment.

Consider a true path $P = \{m_1, m_2, ..., m_n\}$. For every two consecutive points m_i and m_{i+1}, calculate the shortest path in the HRN between those points. Suppose that shortest path is $\{m_i, o_1, o_2, \ldots, o_k, m_{i+1}\}$, possibly containing other HRN points between m_i and m_{i+1}. We add a sample to the travel time distributions of each segment $(m_i, o_1), (o_1, o_2), \ldots, (o_k, m_{i+1})$. Specifically, we divide the total time from m_i to m_{i+1} equally across all the segments of the shortest path. These samples add to any previous empirical distributions of travel times on the path segments (see Algorithm 3).

Input: HRN graph G'(V ',E'), and true paths
Output: Travel time distribution for each HRN segment

forall the *True paths P* **do**
  forall the $(m_i, m_{i+1}) \in P = \{m_1, m_2, ..., m_n\}$ **do**
    sp is the shortest path m_{i-1} to m_i
    ΔT_i is time difference from m_{i-1} to m_i
    forall the *segments* $e_j \in sp = \{m_i \to o_1 \to o_2 \ldots o_k \to m_{i+1}\}$ **do**
      `//` $\mathcal{D}(e_j)$ `is the empirical distribution of travel times for` e_j
      Add $\Delta T_i/(k+1)$ to $\mathcal{D}(e_j)$
    end
  end
end

Algorithm 3: Computing Travel time distributions

5.2 Identifying Intersections for TCDs

The best intersections for installing TCDs are those that would produce the greatest expected time savings for ERVs traversing the transportation system. Two key factors go into identifying these intersections: the time delay on each segment, and the usage frequency of the segment. Both of these parameters contribute to computing the total expected time savings.

To compute expected time savings from installing a TCD on a given intersection, we use the following process. If mean speed of an ERV on a segment is less than the suggested speed limit the segment, we calculate the time that would be saved if the ERV were to travel at the speed limit. We assume ERVs could potentially travel at the speed limit because ERVs enjoy relaxations regarding many traffic regulation like speed limit, red signal etc.

To formally define a procedure for identifying intersection for TCDs, we introduce some additional notation. Let $\mathcal{D}(e_i)$ be the empirical distribution of travel times for an HRN segment e_i. We then define a segment weight $e_i.\texttt{weight}$ as follows:

$$e_i.\texttt{weight} = \sum_{t \in \mathcal{D}(e_i)} (t - \text{EstimatedTime}(e_i)), \tag{5.1}$$

$$\text{EstimatedTime}(e_i) = \frac{\text{Length}(e_i)}{\text{SpeedLim}(e_i)}. \tag{5.2}$$

We use the segment weights to compute vertex weights. The weight $v_i.\texttt{weight}$ on each vertex v_i is calculated by summation of all segment weights $e_i.\texttt{weight}$ for segments adjacent to v_i as given by

$$v_i.\texttt{weight} = \sum_{\forall e_i \text{ adjacent to } v_i} e_i.\texttt{weight}.$$

For example, in Figure 5.1, the weight of v_1 is a summation of segment weights for e_1, e_2, and e_3. This process is summarized in Algorithm 4.

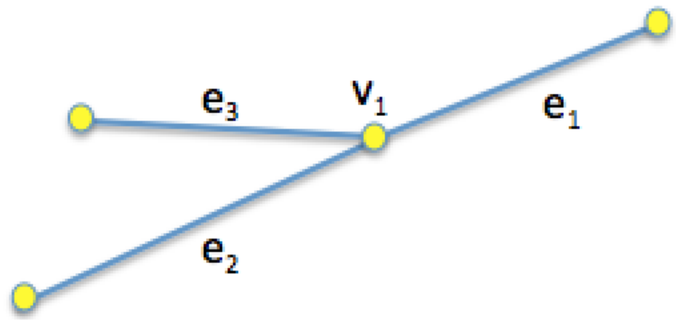

Figure 5.1: Image showing weight calculation for HRN nodes (v_1.`weight` $= \sum_{i=1}^{3} \mathcal{D}(e_i)$)

Input: HRN graph G'(V ',E'), and true paths
Output: Potential locations for TCD installation

forall the $e_i \in E'$ **do**
  // v_t, v_s `are vertices adjacent to` e_i
  // e_i`.weight is calculated using Equation` (5.1)
  v_t.`weight` $=$ v_t.`weight` $+$ e_i.`weight`
  v_s.`weight` $=$ v_s.`weight` $+$ e_i.`weight`
end

Algorithm 4: Computing potential locations for TCD installation

Chapter 6

Results

The GPS data used for testing the algorithm is acquired from the Austin Fire Departments ERVs. The data spans 2 years of incidents starting from January 2013 to December 2014 collected across different types of ERV. Each incident corresponds to response of an emergency call from different locations in Austin. The GPS data is collected only when the ERV is traveling in response to an emergency call. That is the start point of every route is GPS co-ordinates of ERV when an incident is assigned to the ERV, and the end point corresponds to GPS co-ordinates of the incident location. This kind of data is ideal for using our algorithm because it helps to focus on critical response time of ERVs, which emergency service providers try to optimize. By running our algorithm on the entire database, we create files containing HRN line segments, total time spent on each segment, and mean speed for each segment. These files can be loaded into visualization softwares like QGIS.

Figure 6.1 shows Austin road network segments, weighted on total time spent by ERVs attending emergency incidents from January 2013 to December 2014. The more total time ERVs have spent on a road segment, the darker the green shade corresponding to that segment. The yellow dots represent the locations of fire stations in Austin. The dark green color segments near the fire stations confirms our intuition that those routes are used more frequently by the ERVs.

Figure 6.2 shows Austin HRN, weighted on mean speed of ERVs attending emergency incidents from January 2013 to December 2014. With the increase of mean speed on a road segment, its color changes from red to green through yellow shade. The yellow dots represent the locations of fire stations. The red shade on expressway passing through Austin and green shades on express way on outer Austin matches with our intuition. These visualizations serve intuitive checks that the our method work properly.

Figure 6.3 points to a few potential locations for TCD installations. Most of these points are crowded near fire stations; this is likely because those intersections are the ones most frequently used by ERVs. Figure 6.4 represent important intersections, filtering the obvious intersections near the fire stations.

Figure 6.1: Austin HRN, weighted on time spent by ERVs attending emergency incidents from Jan 2013 to Dec 2014. the darker the green shade the more time ERVs have spent on a road segment.The yellow dots represent locations of fire stations.

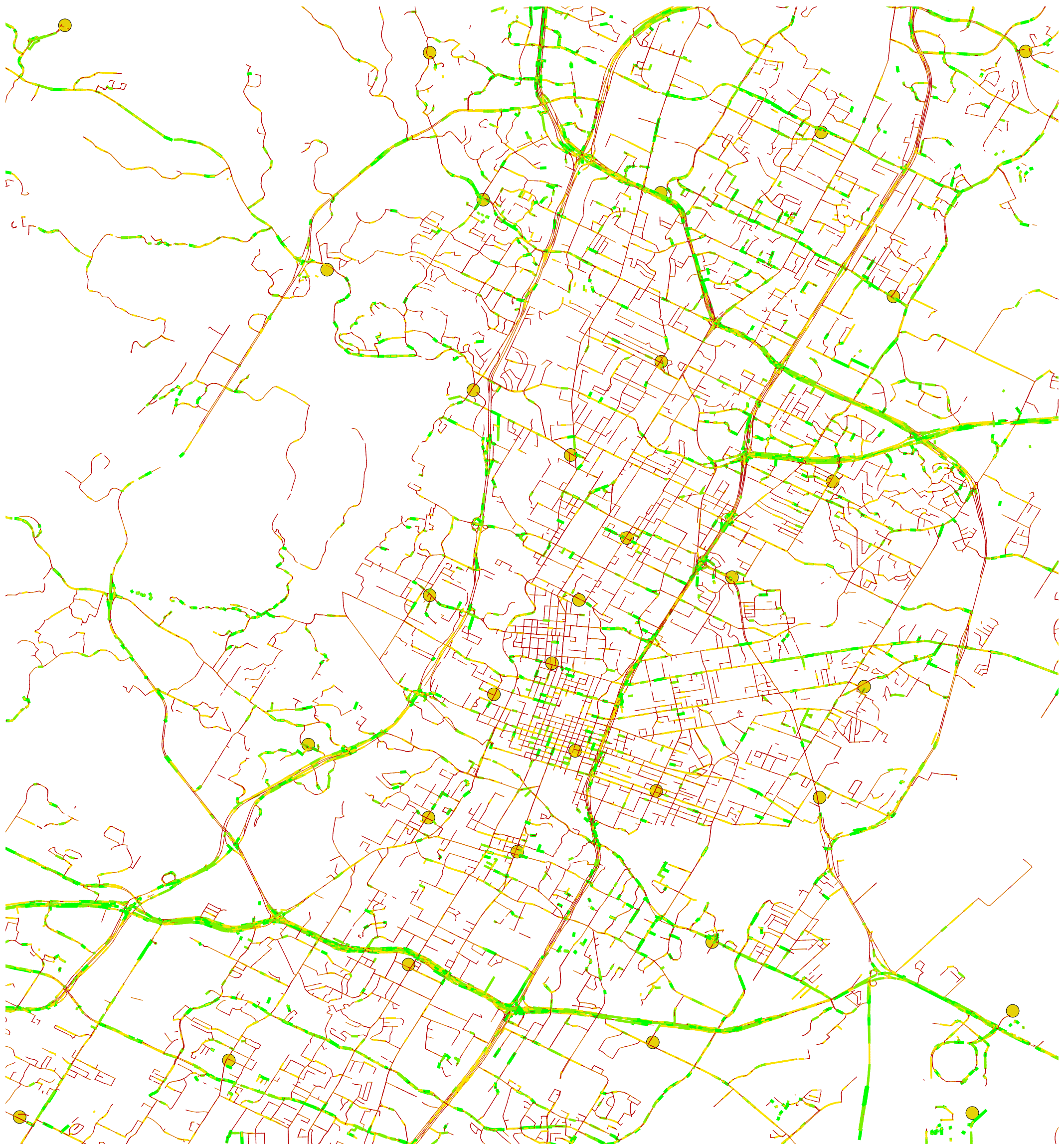

Figure 6.2: Austin HRN, weighted on mean speed of ERVs attending emergency incidents from Jan 2013 to Dec 2014. With the increase of mean speed on a road segment, its color changes from red to green through yellow shade. The yellow dots represent the locations of fire stations.

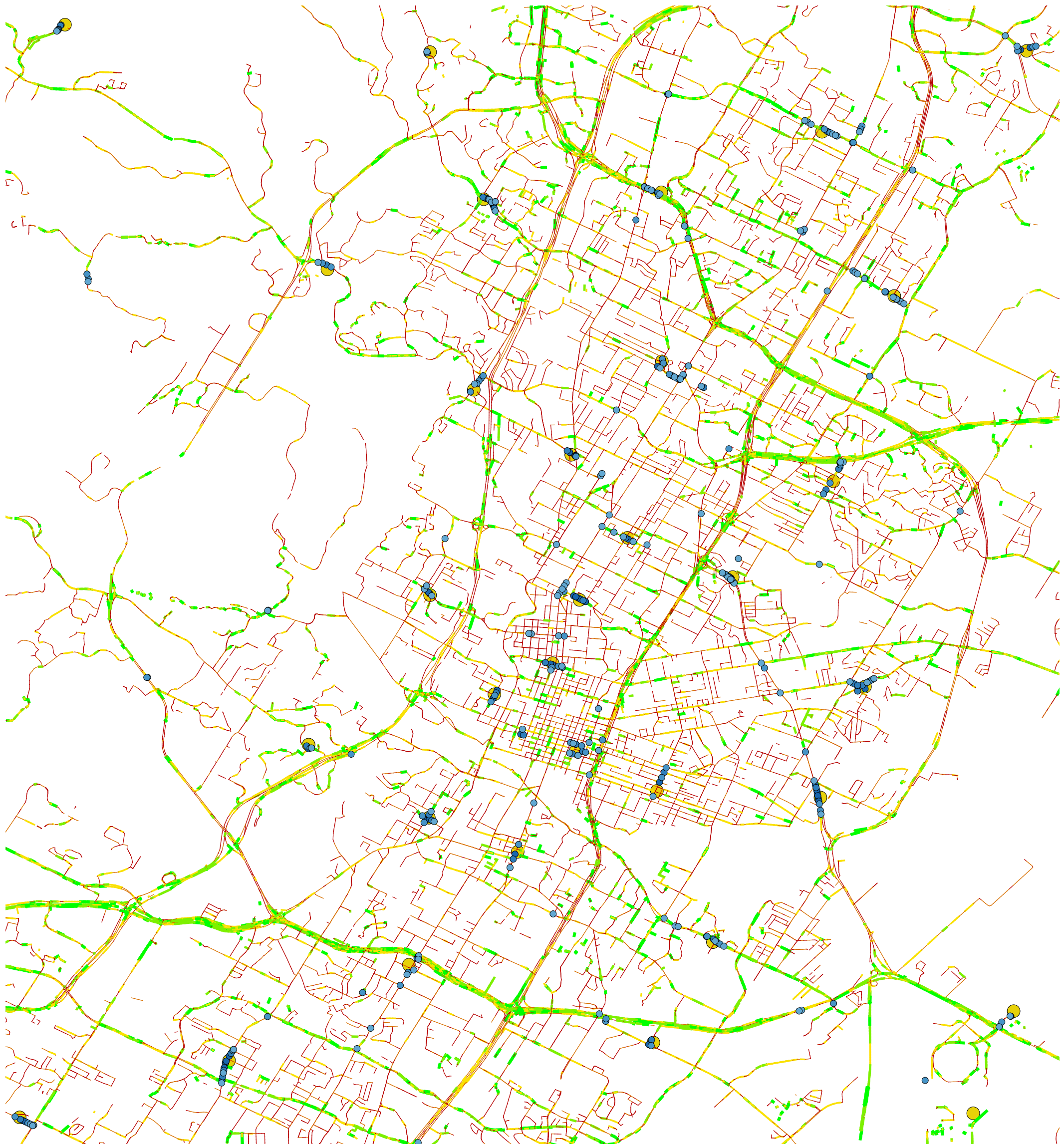

Figure 6.3: Austin HRN from Figure 6.2. With blue dots representing potential TCD installation locations, yellow dots represent the locations of fire stations.

Figure 6.4: Austin HRN from Figure 6.2. With blue dots representing potential TCD installation locations.After filtering intersections close to fire stations represented by yellow dots.

Chapter 7

Conclusion and Future Work

In this thesis, we extend previous map matching methods to be able to derive and analyze speed and travel time distributions across a transportation network. We apply our methodology to data from the Austin Fire Department, consisting of over three hundred thousand response routes. This data can be processed to provide operationally useful information, like identifying intersections for the placement of TCDs.

The algorithms we present leave some room for future improvement. For example, it would be interesting to study how the resolution of the HRN changes the determination of speed and travel time distributions. In addition, we allocate travel time equally to each segment in a true path, which could potentially be improved through a distance-based allocation. Finally, our current methodology for identifying TCD intersections does not take into account correlations in ERV routes. In other words, perhaps only a group of TCDs can make an actual difference in ERV response times. The current analysis assumes benefits at a local intersection level.

The results of our study can be useful in future work. For example, one could use speed and travel time data – filtered appropriately by day and time of day – to route ERVs and achieve potentially faster responses. The data could also be used to identify under-served areas of Austin. For example, it could be used to identify the locations in Austin, where fire response would be the slowest. Finally, this data could be potentially used to inform future resource investment in new fire trucks, selecting a type of truck, fire station locations, and advanced TCD allocation.

Bibliography

[1] US Access Board. Traffic Signal Controllers and APS Technologies. `https://www.access-board.gov/research`, 2016. [Online; accessed 19-April-2016].

[2] Jennifer Dill, Joseph Broach, Kate Deutsch-Burgner, David M Levinson, and Wenyun Tang. Multiday GPS travel behavior data for travel analysis.

[3] Amir Elalouf. Efficient routing of emergency vehicles under uncertain urban traffic conditions. *Journal of Service Science and Management*, 5(3):241, 2012.

[4] Joshua S Greenfeld. Matching GPS observations to locations on a digital map. In *Transportation Research Board 81st Annual Meeting*, 2002.

[5] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and Per-Johan Nordlund. Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, 50(2):425–437, 2002.

[6] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 352–361. ACM, 2009.

[7] Dieter Pfoser and Christian S Jensen. Capturing the uncertainty of moving-object representations. In *Advances in Spatial Databases*, pages 111–131. Springer, 1999.

[8] Mohammed A Quddus, Washington Y Ochieng, and Robert B Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312–328, 2007.

[9] Mohammed A Quddus, Washington Yotto Ochieng, Lin Zhao, and Robert B Noland. A general map matching algorithm for transport telematics applications. *GPS solutions*, 7(3):157–167, 2003.

[10] Prakash Ranjitkar, Takashi Nakatsuji, and Motoki Asano. Performance evaluation of microscopic traffic flow models with test track data. *Transportation Research Record: Journal of the Transportation Research Board*, (1876):90–100, 2004.

[11] Nadine Schuessler and Kay W Axhausen. Map-matching of GPS traces on high-resolution navigation networks using the multiple hypothesis technique (MHT). *Arbeitsberichte Verkehrsund Raumplanung*, 568, 2009.

[12] John JJS Sia, Philip CH Ching, and Prakash Ranjitkar. Travel time study of textAuckland arterial road network using GPS data. In *Australasian Transport Research Forum (ATRF), 32nd*, 2009.

[13] George Taylor, Geoffrey Blewitt, Doerte Steup, Simon Corbett, and Adrijana Car. Road reduction filtering for GPS-GIS navigation. *Transactions in GIS*, 5(3):193–207, 2001.

[14] Christopher E White, David Bernstein, and Alain L Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, 8(1):91–108, 2000.